

PATENT APPLICATION
ON-CHIP SERVICE PROCESSOR FOR TEST AND DEBUG OF
INTEGRATED CIRCUITS

Inventor(s):

Bulent Dervisoglu, residing at
495 Sleeper Avenue
Mountain View, California 94040
a citizen of the United States of America

Laurence H. Cooke, residing at
25399 Spanish Ranch Road
Los Gatos, California 95033
a citizen of the United States of America; and

Vacit Arat, residing at
10699 Magdalena Road
Los Altos Hills, California 94024
a citizen of the United States of America

Entity: Small

0925726.032499

On-Chip Service Processor For Test And Debug of Integrated Circuits

This application claims the benefit under Title 35, United States Code § 119(e) of Provisional Application No. 60/079,316 filed March 25, 1998.

5

BACKGROUND OF THE INVENTION

The present invention is related to the testing and debugging of electronic systems, and, in particular, to on-chip circuits for the test and diagnosis of problems in an integrated circuit.

10

Heretofore, logic analyzer probes have often been used in the testing and debugging of electronic systems. The logic analyzer probes were coupled to the external pins of components of a digital system in order to capture the sequence of signals after a predefined event (or time stamp) occurs. The captured signals can then be examined to verify correct system behavior or, alternatively, to identify the time and the nature of erroneous behavior in the system.

15

Furthermore, in the designs of large electronic systems, separate consoles, or service processors, have often been incorporated into the circuit boards of the system. These separate processors have a number of useful functions, including the control of scan strings in the system; the origination of diagnostic signal probes to run on the system, and so forth. The service processors also have diagnostic and scan debug features, including access to the internal registers and memory within the system. The service processors have also been used to bring-up the main system during its power up phase. All of these functions have been useful to system designers for the design, test and debugging of electronic systems.

20

25

On the other hand, more and more digital systems, or parts of digital systems, are being integrated in a single component. The resulting complexity and lack of observability of an integrated circuit poses serious problems for the test, debug and bring-up stages of the integrated circuit (IC). For example, observation at the IC component pins of the behavior of an IC system is increasingly difficult. The IC component pins may be very far (in terms of logic hierarchy) from the actual points of interest. The extremely high frequency of digital IC operations and the frequency filtering effects of the large capacitance of the external logic analyzer probes, often prevents a logic analyzer from capturing signals reliably and precisely. There is always

30

an uncertainty regarding the accuracy of signals captured by an external logic analyzer compared to the actual signals values within the IC.

To address the problems of the testing of integrated circuits, special features are being included in many IC designs. For example, one standard technique is “scan” whereby, certain internal flip-flops, which are connected to various selected points of the IC, are also connected to form a serial shift register when the IC is configured in a test mode. Straightforward serial shift (i.e., scan) operations are utilized to load the flip-flops with desired values, or to read out their present values reflective of the logic states of the selective IC points. Such ICs require special features to reset the flip-flops (i.e., bring the IC to a known starting state). However, the size of integrated circuits has grown to the point where it has become inefficient and expensive to test and debug ICs using solely conventional scan techniques.

Furthermore, variations of the serial scan technique include the use of so-called “shadow registers.” IC internal signal states are captured in a duplicate copy, i.e., the shadow register, of certain internal registers. The shadow registers are interconnected by a dedicated internal scan chain. A predetermined event can trigger a snapshot of the internal state values in the shadow registers and the dedicated scan chain shifts the captured signal state without affecting the system operation of the IC. However, this approach has several deficiencies. First, only a single snapshot can be captured and shifted out with each trigger event. This greatly hampers debugging the IC since there is not much visibility of the system activity around a point of interest identified by the trigger event. Secondly, the snapshots can be taken only of those signals in registers which have a shadow register counterpart. Since a shadow register effectively doubles the circuitry for the register, this approach is very costly to implement on a large scale in the IC.

Another test and debug design for ICs is found in a standard, the IEEE 1149.1 Test Access Port and Boundary-Scan Architecture, which prescribes a test controller which responds to a set of predetermined instructions and an instruction register which holds the present instruction which the controller executes. Each instruction is first loaded into the instruction register from a source outside the IC and then that instruction is executed by the controller. While having some advantages of versatility and speed, the standard still binds test and debug procedures to the world external to the IC and thus, limits its performance.

09275726-032499
The present invention recognizes that while the advances in IC technology have helped to create the problems of testing and debugging an IC, the advances also point the way toward solving these problems. In accordance with the present invention, special on-chip circuits are used to observe the internal workings of an IC. These circuits
5 operate at internal IC clock rates so that the limitations of the frequency of signals at the IC input and output (I/O) boundary are avoided. Many more points in the IC system are accessed than is feasible with conventional external test and debug processors. Thus the present invention offers advantages which exceed the straight-forward savings in chip space due to miniaturization. Additionally, the present invention reduces the amount of
10 test logic which might have been required elsewhere on the chip.

The present invention also permits the coupling of probes to internal IC points. The points may be selected from a larger number of internal points that may be observed with an external logic analyzer. Besides the greater observability of the internal operations of the IC, the present invention also improves the accuracy of the observations,
15 as compared to an external logic analyzer.

SUMMARY OF THE INVENTION

To achieve these ends, the present invention provides for an integrated circuit logic blocks, a control unit, a memory associated with the control unit and a
20 plurality of scan lines. The memory holds instructions for the control unit to perform test and debug operations of the logic blocks. The scan lines are responsive to the control unit for loading test signals for the logic blocks and retrieving test signal results from the logic blocks. The test signals and the test signal results are stored in the memory so that the loading and retrieving operations are performed at one or more clock signal rates internal
25 to the integrated circuit. The integrated circuit also has a plurality of probe lines which are responsive to the control unit for carrying system operation signals at predetermined probe points of the logic blocks. The system operation signals are also stored in the memory so that the system operation signals are retrieved at one or more clock signal rates internal to the integrated circuit.

30 The present invention also provides for an integrated circuit which has an interface for coupling to an external diagnostic processor, a unit responsive to instructions from the external diagnostics processor, a plurality of probe lines coupled to the unit, and a memory coupled to the unit and to the interface. In response to the unit, the probe lines carry sequential of sets of system operation signals at predetermined probe points of the

integrated circuit and the system operation signals are stored in the memory at one or more clock signal rates internal to the integrated circuit. The system operation signals are retrieved from the memory through the interface to the external diagnostic processor at one or more clock signal rates external to the integrated circuit. This allows the external

5 diagnostics processor to process the captured system operation signals.

The present invention further provides for a method of operating an integrated circuit which has logic blocks, a control unit, a memory and a plurality of scan lines of the logic blocks. The memory is loaded with test signals and instructions for the control unit and the scan lines responsive to the control unit are loaded with the test

10 signals for the logic blocks at one or more clock signal rates internal to the integrated circuit. The logic blocks are then operated at one or more clock signal rates internal to the integrated circuit and the resulting test signal results are retrieved from the logic blocks along the scan lines at one or more clock signal rates internal to the integrated circuit. The test signal results are stored in the memory at one or more clock signal rates

15 internal to the integrated circuit; and the stored test results signals are processed in the control unit responsive to the stored instructions in the memory to perform test and debug operations of the logic blocks.

BRIEF DESCRIPTION OF THE DRAWINGS

20 Figure 1a shows a high-level diagram of an exemplary large and complex integrated circuit. Figure 1b shows the Fig. 1a integrated circuit with a Service Processor Unit (SPU), according to one embodiment of the present invention;

Figure 2 illustrates one embodiment for the architecture for the SPU of Figure 1b;

25 Figure 3a illustrates the coupling between test wrappers, scan strings, probe strings and trigger signal lines to a test bus; Figure 3b is a circuit diagram of a test bus connector of Figure 3a; Figure 3c is an exemplary connection of multiple test bus connectors;

Figure 4a is a circuit diagram of a block input/output connector for test

30 wrappers for observing test points outside a block along a boundary-scan chain (for example, IEEE 1149.1 standard Test Access Port and Boundary Scan Architecture); Figure 4b is a circuit diagram of a block scan connector for scan strings for observing test points inside a block along a scan chain;

Figure 5 is a circuit diagram of a scan flip-flop in the Fig. 4b circuit diagram;

Figure 6a is a circuit which generates an out-of-range detection probe signals for analog probes in Figure 2; Figures 6b and 6c are the transistor-level circuits of
5 inverters in Figure 6a.;

Figure 7 is a circuit which generates ground-bounce detection probe signals for analog probes;

Figure 8 is a block diagram of a Built In Self-Test (BIST) engine of the Figure 2 SPU;

10 Figure 9a is a block diagram of an input aligner portion of Analysis Engine of the Figure 2 SPU; Figure 9b is a detail of the Figure 9a Analysis Engine's input aligner; Figure 9c is a block diagram of the Analysis Engine's memory addressing structure; Figure 9d is a block diagram of the trigger logic portion of the Analysis Engine; and

15 Figure 10 is a block diagram of another embodiment of the Analysis Engine's memory addressing structure;

Figure 11 shows a probe string connection of probe points to the buffer memory using logic analyzer channels that are implemented with probe storage elements (PSE);

20 Figure 12 shows an alternative probe string connection with improved multiplexed PSEs which combine probe selection and data capture functions; and

Figure 13 is a block diagram of the improved PSE of Figure 12.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

25 General Organization of the Present Invention

In accordance with the present invention, a Service Processor Unit (SPU) is incorporated within an integrated circuit. Besides addressing the problems of testing and debugging the IC, the availability of a programmable unit, such as the SPU, which may load or unload the state variables into and from the user-definable logic in an IC,
30 greatly simplifies the problem of resetting the IC and observing its current state. The SPU is implemented in the form of a basic stored-program control unit, such as a microprocessor, with a predefined instruction set, a number of extended function units (EFUs), program, data, and scratch pad memories, plus an input/output circuit for loading and unloading the SPU memories with data/programs from the outside world.

09275726.032499

This allows the SPU to be programmed to execute a control program which interacts with the various extended functional units to control various test and debug related activities on the IC.

Each EFU is designed to control a specific test or debug feature and the
5 EFU provides the control unit a general, programmable access to that feature. For example, one EFU may be designed to control the execution of serial shift operations along some or all of the internal scan chains of the IC. The other EFUs may be enabled to interact with the scan chains, such as a predetermined algorithm to provide a Built-In Self-Test (BIST) for an embedded Random Access Memory (RAM) block. The existing
10 scan chains load and unload the BIST patterns and results to/from the RAM block. The EFUs provide the control unit with a straight forward, programmable means for controlling the functions of the EFU such that knowledge of low level details of the scan or BIST functions become unnecessary.

With its program and data memories, the SPU acts autonomously once its
15 program memory has been loaded with the desired instruction sequence. The SPU's program memory may be loaded with the desired program instructions through the SPU's interface to the external environment. Alternatively, the instructions may be stored in an on-chip Read Only Memory (ROM) that has been provided to work as the SPU's program memory.

20 In one embodiment of the present invention, an EFU carries out certain functions of a logic analyzer. A logic analyzer captures and stores signal state values in a digital system following the occurrence of a pre-defined event. The logic analyzer then analyzes the captured data and displays the results for perusal. With the present invention, the capture and storage functions are incorporated into the IC. The EFU which
25 implements these functions captures and stores not a single snapshot but a sequence (i.e., history) of signal values using logic probes which are selectively coupled to desired points in the IC logic circuits. The logic analyzer EFU is configurable to select the location, number and sequential depth of signal channels from a predetermined set of choices. Thus, each logic analyzer channel may be selectively coupled to more than one
30 predetermined capture point by programming the control unit and hence, the EFU. A solution is provided for capturing the history of signal values at the internal points of the IC without having to provide each one of these points with their shadow register counterpart. The captured data are stored in an on-chip Random Access Memory (RAM). Transportation of the captured data out of the IC is performed later for analysis by an

external computer which can reformat and display as required for diagnostics. The present invention has the benefit of enhanced data accuracy with minimal cost overhead by separating the signal capture/storage function of a logic analyzer into the IC.

Two different types of logic probes may be used with the logic analyzer EFU. One type of logic probe, termed the digital probe, captures sequences of digital signals from internal points of the IC. Digital signal values flow from the internal capture point to a logic analyzer channel through the digital probe. In its simplest form each digital probe has at least two input ports, a selection means and an output port that is directly coupled to a logic analyzer channel. Digital probes may also be constructed from a series of internal storage elements (i.e., flip-flops or latches) to form a pipeline to move the data from the capture points towards the logic analyzer channels. In this case, the movement of the data along the digital probe flip-flops is synchronized with an on-chip clock signal. Since the clock frequency also defines the maximum capture rate, the particular clock signal is selected based on the maximum desired capture rate. The digital probes used for the logic analyzer EFU operate with the same electrical and timing characteristics of the native signals of the IC. The digital probes are implemented in the same technology, with the same functional logic circuitry, and under the same clock timing, as the rest of the IC. Signals are ^{for} therefor captured and propagated along the digital probes in exactly the same way as they are operated upon by the functional circuitry of the IC. This assures much greater accuracy of signal states captured by the digital probes. In contrast, logic probes used with an external logic analyzer must use trigger events and signal values that are visible external to the IC. The captured signal values may differ significantly from the original (internal) values.

The logic analyzer EFU may use a second type of logic probe, termed an analog probe, which captures signal events representing the detection of signal integrity conditions, such as ground bounce. Desired signal observation points are coupled to analog detection circuits which produce digital signals when particular signal conditions are detected. The analog probe ^{reads} records these digital signal states in the logic analyzer EFU.

The benefits of the logic analyzer EFU are such that for certain ICs, only the EFU portion of the SPU is implemented on the IC. In this alternate embodiment of the present invention, the digital and analog probes are selectively enabled by a scan-chain which allows specific control signals to be loaded into these probe circuits. The scan chain also carries other control signals to be loaded into a trigger circuit which starts

and stops the data capture operations. Once the desired data has been captured into an on-chip RAM, the data is transported outside the IC for subsequent analysis and display.

Implementations of the Present Invention

5 As a starting point, Fig. 1a is a diagram of an exemplary integrated circuit. The IC 100 is complex having a host processor connected by a system bus to various circuit blocks, including a third party core and other blocks adapted to the application of the IC. The IC also has a peripheral bus which is connected to the system bus by a bridge. The peripheral bus is connected to other functional blocks, such as a user-
10 developed core and so on.

A preferred embodiment of the present invention to test and debug the complex IC of Figure 1a is shown in Figure 1b. Added to the IC 100 is a Service Processor Unit (SPU) 101 which is coupled to the IC system bus 105 and an added test bus 104. Connected to the test bus 104 are test wrappers 102 which provide test
15 communication channels into selected blocks 106. More details of the test bus 104 and test wrappers 102 are provided below. The SPU 101 provides a connection for an external diagnostics console 103 to view and test the internal workings of the IC 100.

As shown in Figure 2, the SPU 101 has several extended function units (EFUs), including a control unit, such as a microprocessor 211, a buffer memory unit 218,
20 an analysis engine 215, a scan control unit 222, an interrupt handler 221, which is further connected to a range check unit 220, a system bus interface 214, a test bus interface 213 and a built-in self test (BIST) engine 212, which are all interconnected by a processor bus 219. The various EFUs are coupled to the processor bus 219 in any desired combination and order. To provide communication between the external world and the SPU 101, the
25 bus 219 is also connected to a serial input/output (SIO) interface 210, a parallel input/output interface (PIO) 216, and a test access port (TAP) 217. For example, the coupling between the IC 100 and the external diagnostics console 103, typically implemented using another computer, uses the TAP 217, the SIO interface 210 or the PIO interface 216.

30 Analog probe lines 201 are connected to the range check unit 220 which processes their values to detect out-of-range conditions which are then signaled to the interrupt handler 221. The interrupt handler 221 also receives signals from trigger event lines 204 directly or from test bus 104 by way of test bus connections 203 to the interrupt handler 221. The signals on the trigger event lines 204 or test connections 203 are used to

09275726.032499

capture signal state values when predetermined (i.e., triggering) events occur. The interrupt handler 221 passes the captured values to the analysis engine 215. The test bus 104 is further coupled to test wrappers 102, which are individually wrapped around a number of predetermined blocks 106 on the IC 100. Each test wrapper 102 accesses the input and output signals of a block 106. The test bus 104 is also connected to scan string lines 403, which are connected to internal elements of a block 106.

As shown in Figure 3a, the test bus 104 forms a unidirectional loop with test bus connectors 401 selectively transferring data between the test bus 104 and a test wrapper 102. The test bus 104 is made up of multiple bit lines, where the number of the bits is determined by the requirements of the test system. Through test bus connector 401, the test bus 104 is selectively connected to test wrappers 102, scan string lines 403, probe string lines 402 and trigger lines 204.

A test bus connector 401 which handles a one bit connection between the test bus 104 and a test wrapper 102 is illustrated in Figure 3b. A first multiplexer 421 has one of its input terminals connected to one of the lines of the test bus 104. The other input terminal is connected to a signal line of the test wrapper 102. The output terminal of the multiplexer 421 is connected to an input terminal of a flip-flop 426 and to an input terminal of a second multiplexer 422, which has a second input terminal connected to the output terminal of the flip-flop 426. The output terminal of the flip-flop 426 is also connected to the line of the test wrapper 102, which is also in the form of a unidirectional loop. The multiplexer 421 selects either the data from the test bus 104 or the test wrapper 102; the second multiplexer 422 selects between the data selected by the first multiplexer 421 or the data captured in the flip-flop 426 to place back onto the test bus 104. These selections are done under the control of SPU 101. The test bus connector 401 is also used for coupling a trigger line 204, probe string line 402 or scan string line 403 to a test bus 104 by connecting the desired signal line in place of the line of the test wrapper 102 port as shown in Figure 3b.

Figure 3c shows an embodiment of coupling a trigger line 204, probe string 402, test wrapper 102 and scan string line 403 to three lines of the test bus 104.

Other possible configurations for the couplings include coupling the test wrapper 102 and scan string 403 onto separate lines of the test bus 104.

A test wrapper 102 is formed by serially connecting block I/O connector circuits 310. One such circuit 310, which couples an input or output signal of a block 106 to the test wrapper 102, is illustrated in Fig. 4a. The connector circuit 310 has a scan-in

terminal 304 and a scan-out terminal 306. The scan-in terminal 304 of one circuit 301 is connected to the scan-out terminal 306 of another circuit 301 to form the serial chain of a test wrapper 102. The connector circuit 310 also has a data-in terminal 302 and a data-out terminal 307 which provide an interstitial connection between a block 106 and the rest of the IC 100. In the normal operation of the IC, the connector circuit 310 provides a simple path between the block 106 and the rest of the IC 100. If the connector circuit 310 is to provide an input signal to the block 106 during test operations, the data out terminal 307 is connected to the block 106 and the data in terminal is connected to the rest of the IC 100. If the block I/O connector circuit 310 is to receive an output signal from the block 106 during test operations, the data-out terminal 307 is connected to the rest of the IC 100 and the data-in terminal is connected to the block 106. The connector circuit 310 also has a probe-in terminal 303 and a probe-out terminal 305 which provide a path for probe signals from selected portions of the block 106 through the connector circuit 310 to observe operations in the block 106.

The elements of the connector circuit 310 include a scan flip-flop 301 and two multiplexers 308 and 309. The data-in terminal 302 and the scan-in terminal 304 form the inputs to the flip-flop 301. The output from the flip-flop 301 include the scan out terminal 306 and one input to the multiplexer 308 having an output which forms the data-out terminal 307. The second input to the multiplexer 308 is connected to the data-in terminal 302, which is also connected to one input to the multiplexer 309. The probe-in terminal 303 forms a second input to the multiplexer 309 whose output forms the probe-out terminal 305. The control input of the multiplexer 309 is the output of the scan flip-flop 301 (and is connected to one input of the multiplexer 308). The control input of the multiplexer 308 is a test control line 300 from the control unit 311 of the SPU 101. The control signal on the line 300 selects whether the functional signal at data-in terminal 302 or the signal held in the scan flip-flop 301 is passed onto the data-out terminal 307. When the control signal of the line 300 signal is not-asserted, i.e., normal mode, there is normal operational signal flow between the data-in terminal 302 and the data-out terminal 307. On the other hand, when the control signal on the line 300 is in asserted state, i.e., test mode, the current state of the scan flip-flop 301 is passed onto the data-out terminal 307; the data-in terminal 302 and the data-out terminal 307 are isolated from one another. The state stored in the scan flip-flop 301 also controls whether the signal at the data-in terminal 302 or the probe-in terminal 303 is passed onto the probe-out terminal 305. In this manner, data from another probe point which is connected to the probe-in terminal

303 are selectively passed onto the probe-out terminal 305. The signal state in the scan flip-flop 301 value is controlled and observed using regular scan operations of the test wrapper 102 through the scan-in and scan-out terminals 304 and 306. Of course, if observation of an input or output signal of the block 106 by a probe string 402 is not required, the multiplexer 309 can be eliminated from the circuit 310.

A scan string 403 is formed by serially connecting block scan connector circuits 320. One such circuit 320, which couples an internal element of a block 106 to the scan string 403, is illustrated in Fig. 4b. The connector circuit 320 has a scan-in terminal 314 and a scan-out terminal 316. The scan-in terminal 314 of one connector circuit is connected to the scan-out terminal 316 of another connector circuit 320 to form a serial scan string 403. The block scan connector circuit 320 also has a data-in terminal 312 and a data-out terminal 317 which provide an interstitial connection between internal elements of the block 106. In the normal operation of the IC 100, the connector circuit 320 is a simple path between the internal elements in the block 106. The connector circuit 320 also has a probe-in terminal 313 and a probe-out terminal 315 which provide a path for probe signals from selected portions of the block 106 through the connector circuit 320 to observe operations in the block 106.

The block scan connector circuit 320 has a scan flip-flop 311 and a multiplexer 319. The data-in terminal 312 and the scan-in terminal 314 form the inputs to the scan flip-flop 311. The output from the flip-flop 311 include the scan out terminal 316 and the data-out terminal 317. The data-in terminal 302 is also connected to one input to the multiplexer 319. The probe-in terminal 313 forms a second input to the multiplexer 319 whose output forms the probe-out terminal 315. A special circuit is used for the scan flip-flop 311 (and the flip-flop 301 of Figure 4a). The circuit, which is shown in Figure 5 and is found in previous IC scan designs, has separate scan-slave and data-slave sections. The separation allows a state signal which has been scanned into the scan flip-flop 311 to remain unaffected by functional clock pulses that cause the flip-flop 311 to capture signals on the data in terminal 312 so that they appear in the data-slave section and on the data out terminal 317. The connector circuit 320 acts as a simple conduit for signals within the block 106. At the same time, the previously scanned-in signal, which appears in the scan-slave section, selects whether signals at the data in terminal 312 or the output from another probe point which has been connected to the probe-in terminal 313 is to be passed onto the probe-out terminal 315. A probe string 402 is created. Of course, if an

internal scan string 403 need not be connected to a probe string 402, the multiplexer 319 can be eliminated from the circuit 320.

A probe string 402 is formed by serially connecting the probe-in terminal of a connector circuit 310 and 320 to the probe-out terminal of another connector circuit 310 and 320. The probe string 402 typically has a set of selectively connected probe points. However, only one probe point along each probe string 402 may be actively probed at any given time. Thus the IC designer selects the probe points which are to be connected along the same probe string 402 and determines the total number of probe strings 402 that are to be connected to the individual bits of the test bus 104. This structure allows the IC designer great flexibility to optimize the number of test bus 104 lines with respect to the number of simultaneously observable probe points in the IC.

The probes described above are digital probes. Two analog probes are illustrated in Figures 6a, 6b, 6c and 7. The range check unit 220 receives inputs from the analog probes that comprise signals on a threshold check line 600 and a ground bounce line 700. The unit transmits these signals to the SPU 101. Figures 6a, 6b and 6c show the circuit which generate the signal for the threshold check line 600. The circuit is used for detecting extended intermediate voltage levels. Such voltage levels are most likely to occur on an on-chip bus which is in contention among multiple circuit drivers. The analog probe has two inverters 601 and 602, which are both coupled to an Exclusive-NOR logic gate circuit. Figure 6b is a transistor diagram depicting the low threshold inverter 601, and Figure 6c is a transistor diagram depicting the high threshold inverter 602. These inverters 601 and 602 exhibit switching properties characteristic of a very low internal voltage, and a very high internal voltage device, respectively. Normally, the circuit in Figure 6a has a logic one (1) output level, but during transitions of the input signal, the outputs of inverters 601 and 602 may remain in opposite states for a period sufficient to cause the circuit to go to a logic zero (0) output level before returning to the logic one (1) output level. This negative pulse can be captured by the SPU 101.

Figure 7 shows a schematic diagram of a ground bounce detector circuit which generates the signals for a ground bounce line 700. In this circuit, a quiet (and true) ground terminal 701 is connected to an N-channel transistor 702, which gate is driven by a local ground connection terminal 703. A periodic clock on a Reset terminal 706, which is controlled from the range check 220, clears a pair of NAND gates configured as a SR latch 704, and charges a capacitor 705 having one terminal connected to the Set input of the SR latch. The second terminal of the capacitor 705 is connected to

09275726-032499

the quiet ground terminal 701. The N-channel transistor 702 which is gated by the local ground discharges the Set line of the SR latch 704, which flips the state of the SR latch 704 if the local ground falls above threshold. For example, a ground spike on the local ground may drive the local ground above threshold. The frequency and duty cycle of the Reset signal determines the magnitude and duration of a ground spike on the local ground to trigger the probe. A variety of frequencies and duty cycles are created by the range check 220 to determine the severity of ground spikes. When the probe is triggered, the probe produces a negative (0) value until reset by the Reset signal on the terminal 706.

Returning to the components of the SPU 101, Figure 8 is a preferred embodiment of the BIST engine 212. A polynomial register 711 identifies the bits in a linear feedback shift register (LSFR) 714 which are used to form an Exclusive-OR (XOR) function which generates pseudo-random values. The polynomial register 711 is set by the microprocessor 211, which also initializes contents of the LSFR 714. The output of the LSFR 717 is connected to the inputs of a multiplexer 715 which also receives the outputs of a mask shift register 712 and a pattern shift register 713. The output of the multiplexer 715 is an input to the LSFR 714. The mask shift register 712 identifies the bit positions whose values are selected from predetermined bit patterns in mask shift register 713 versus the bit positions which receive the pseudo-random values generated by the LSFR 714. The output of the multiplexer 715 is a combination of built-in-self-test and functional scan vectors. These features are useful because random vectors work well only when the controls allow the random vectors to exercise most of the IC section under test. If there are more than a few control lines, the probability of properly exercising the logic under test with random vectors is very low. These features also allow the SPU 101 to generate regularly repeating patterns; for example, periodic patterns that may be useful in a memory test may be generated by the SPU 101 that may output the data to the section of logic under test via the test bus or the system bus, whichever has been provided with a connection to the SPU 101.

Another EFU of the SPU 101 is the analysis engine 215. Figure 9a shows an embodiment of the analysis engine 215 which, under the control of the microprocessor 211, captures logic signals from the test bus 104. This is achieved by first setting either the scan flip-flops 301 of the block I/O connector circuits 310 (Figure 4a) or the scan flip-flops 311 of the block scan connector circuit 320 (Figure 4b) so that a boundary connection or an internal point connection of the target block 106 is selected for probing, respectively. Next, all flip-flops along the same probe string 402 are programmed (by the

SPU 101) so that only signals from the selected probe point are allowed to flow through the probe string 402 and arrive at the test bus connector 401. The multiplexer 421 and the multiplexer 422 in the test bus connector 401 (Figure 3a) are controlled by the SPU 101 so that the signals on the probe string 402 are passed along to the test bus 104. Finally, all remaining test bus connector circuits 401 along the same bit line of the test bus 104 are controlled by the SPU 101 so that they pass the probe signals along test bus 104. This allows the selected probe signal to arrive at the analysis engine 215 where it is captured for subsequent off-line analysis. The input terminals of a plurality of flip-flops 805, one for each bit line of the test bus 104, form the input port 802 of the analysis engine 215. A digital phase locked loop (PLL) 802 has selectable clock outputs 803 to each flip-flop 805 to tune when the data from each probe point is to be captured. The output terminal of each flip-flop 905 is connected to the input terminal of a variable First-In-First-Out shift register (FIFO) 804.

Figure 9b shows the circuit details of each variable First-In-First-Out shift register (FIFO) 804, each having a number of serially-connected register stages 812. Each register stage 812 has a multiplexer which, under control of a decoder 811, selects between the signal held in a flip-flop of that stage or the incoming signal to the stage to place on the stage's output terminal. The shift depth of each variable FIFO 804 is programmable by the SPU 101 by setting a count register 810 for each bit feeding the analysis engine 215. The value in the count register 810 is decoded by the decoder 811. The result controls the number of register stages 812 which are bypassed. This compensates for the path delay differences among the different probe points by realigning capture times of signals captured in the analysis engine 215.

The analysis engine 215 also has trigger logic which control the capture of data. Figures 9c and 9d show sections of the trigger logic, a programmable circuit which detects one or more events to stop the analysis engine 215 from capturing new data. The data that has been captured up to that point is preserved in the buffer memory 218 of the SPU 101. The buffer memory 218 resides in the same address space as the RAM used by the SPU 101 but may be mapped to use high memory space in order to prevent interference with the instructions and data stored in low memory space. When the analysis-engine 215 collects data, it may be allowed to write over old data, keeping only as many most-recent cycles of data as the buffer memory 218 can hold. The size of the buffer memory 218 for the analysis engine 215 is determined by the designer of the IC.

09275726 032499 6642E0 924260

The trigger logic has a start address counter 820 and a stop counter 821, which are shown in Figure 9c. These counters are loaded by the microprocessor 211. The trigger circuit also has an address counter 822 which is designed to overflow at the highest memory address of the buffer memory 218. At that point the start address is
5 reloaded with the beginning address of the high memory space which is reserved for the buffer 218. This converts a random access memory into a FIFO register. The stop counter 821 decrements when a latched trigger signal line 824 becomes set. Subsequently the analysis engine 215 collects data into the buffer memory 218 from the variable FIFOs 804 for as many cycles as defined by the value loaded into the stop counter 821. The
10 system IC designer uses the buffer memory size and the value in the stop counter 821 as two parameters to control the amount of data collected before and after an event has been detected.

Also part of the trigger logic is a circuit which generates the triggering signals on the trigger signal line 824. As shown in Figure 9d, the generating circuit is
15 structured to form Boolean AND-OR logic 831 out of individually selectable terms 832. The terms 832 are fed from a polarity programming logic circuit 833 that accepts individual trigger variables, Probe 1 through Probe N. In addition, the true or the complemented value for the output function can be selected through a final level circuit 830. In one embodiment (shown in Figure 9d), the result is also shifted into three
20 successive flip-flops 834. Each of the flip-flops 834 drives one input of each of a plurality of multiplexers 835. The other inputs of the multiplexers 835 are set to a logic one (1) level. Each multiplexer 835 is individually controlled through programmable bits and the multiplexer outputs are logically ANDed together to form a signal, $T[i]$, which represents the presence of the trigger condition over four consecutive clock periods. The
25 output from the AND gate 836 is passed to an AND gate 837 with inputs from the corresponding AND gates 836 of duplicate circuits that produce $T[0]$, $T[1]$, through $T[n]$ signals. The output of AND gate 837 is stored in a latch 838 to form the latched trigger signal on the line 824. Once the signal is set, the latched trigger signal maintains its value until it is reset through reprogramming by the microprocessor 211. In other
30 embodiments, there may be more or fewer latches, and additional logic to make adjustments to the phases (i.e. the relative clock cycle when signal is received) of the individual signals.

Another embodiment of the trigger logic is shown in Figure 10. This embodiment provides for the capability of reversing the data capturing function of the

analysis engine 215 from continually capturing new data until the trigger detected, to not capturing any data until a trigger is received. In the latter case, each time a trigger signal on the line 824 is received, the analysis engine 215 captures new data for a preprogrammed number of cycles and then stops until the next latched signal on the line 824 is received. To enable this mode of operation, the trigger circuit shown in Figure 10 causes the previous trigger condition to be cleared so that it may be recognized again. This mode is very useful since it enables the capture of signals around (i.e., before and after) multiple occurrences of trigger conditions. The buffer memory 218 is utilized more efficiently as the storage of unwanted cycles of data between the trigger points is not required. It is also possible to program the trigger logic so it uses an externally generated trigger condition 902 in place of an internally programmed event.

Program instructions and initial data values for executing programs to implement the functions of the SPU 101 are loaded from the diagnostics console 103 (see Figure 1b) into the buffer memory 218 of the SPU 101. Some of these programs may access the system bus 105 or the test bus 104. A program can control which test wrapper 102 is accessed by using the test bus interface 213 in order to set control signals on the test bus 104. This allows the SPU 101 to read data from a test wrapper 102 via the test bus 104 into the buffer memory 218 and then send said data out to the diagnostic console 103. Typically, a separate program executed on the diagnostic console 103 displays this information in a human readable format as may be appropriate for the given application.

Programs executed by the SPU 101 can also read data from the diagnostics console 103 via the SIO interface 210 or TAP interface 217, as shown in Figure 2b, and write data out to individual scan flip-flops on the test wrappers 102 via the test bus 104. Significant processing, for example, expansion, compaction, or intermediate storage of data can be done by the SPU 101 utilizing the buffer memory 218. In other embodiments, control functions may be supplied directly from the TAP interface 217 or SIO interface 210 to the analysis engine 215 or BIST engine 212, via the processor bus 219 without involving the microprocessor 211. The SPU 101 may be coupled to either the system bus 205, or a separate test bus 104, or both. The coupling to the diagnostics console 103 may be via the TAP interface 217 or the SIO interface 210. The test bus 104 may be coupled to one or more test wrappers 102.

Another embodiment of the invention is defined in which the SPU 101 does not include an embedded microprocessor 211. In this case, the analysis engine 215 and the BIST engine 212 can access the buffer memory 218 and system bus interface 214

directly, following instructions received from the external diagnostics console 103. In this case, the loading of the configuration information and transfer of data to and from the analysis engine 215 is controlled using hardwired control signals. In this embodiment, the analysis engine 215 is implemented in the form of an on-chip logic analyzer (OLA) which captures sequential snapshots of sets of signals. The selected signals form the digital probes 202. The selections are achieved by coupling the signals for digital probes 202 to the channels of the analysis engine 215 and turning-on enabling circuits, if provided, to allow the signals on the digital probes 202 value to be captured onto channels of the logic analyzer 215. As shown in Figure 11, the channels of the logic analyzer 215 are formed from probe storage elements (PSE) 1000 to form a distributed serial shift register which acts as a pipeline to move data captured at a probe point towards the end of the logic analyzer channel where the data are stored in buffer memory 218. Each channel of the analysis engine 215 contains zero or more number of PSEs 1000 which are clocked by a common periodic clock signal labeled "Cf" on a clock signal line 1001. The clock signal is chosen (at design time) from among the fastest frequency of clock signals which are used in generating source signals to be captured by the probes. This way all signals captured on the analysis engine 215 channels arrive at the end of the channels after a fixed, predetermined number of clock cycles so that their cycle relationship to one another is preserved, regardless of the length (i.e., number of bits) of the individual channels of analysis engine 215 .

Subsequently, after the captured data has been transported to the external diagnostics console 103. software processes use the number of PSEs 1000 on each channel of the analysis engine 215 to align the data with respect to one another. The lengths (i.e. number of bits) of the serial shift registers on the individual channels of the analysis engine 215 are determined at design time so that signal delays due to physical distances among the PSEs 1000 are sufficiently short to allow data to be shifted between consecutive bits of the shift registers in a single clock cycle. If necessary, the number of stages of the shift registers may be increased to satisfy this condition. Each channel of the analysis engine 215 is coupled to a different data input port of the buffer memory 218. The collective data applied to the ports of the buffer memory 218 is written to an address in memory which is identified by a common address register 822 that advances under control of the periodic clock signal "Cf" on the line 1001.

Figure 12 shows a preferred embodiment of a channel of the OLA 215 which uses multiplexed PSEs 100 to combine the selection of probe points and pipelining

captured data into a single, efficient design. This enables the coupling one PSE 1000 to two probe points or another PSE 1000. Scan operations shift a control signal into the PSE 1000 to program itself to select one or the other of its input ports.

The details of a multiplexed PSE is shown in Figure 13. The PSE 1000, 5 illustrated by a dotted line, is connected to a multiplexer 1108 which has two input terminals connected to two input probe paths, P1 and P2, for the logic analyzer channels. Besides the probe clock signal line 1001, which carries the Cf signal, the PSE 1000 is connected to a first scan clock signal line 1101, which carries an A_clk signal, a second scan clock signal line 1102, which carries a B_clk signal, and a scan control line 1103, 10 which carries a Scan_mode signal. The PSE 1000 has three latches 1105, 1106 and 1107. The output terminal of the latch 1105 is connected to one input terminal of the latch 1106 and to one input terminal of the latch 1107. One input terminal of the latch 1105 is connected to the output terminal of the multiplexer 1108 and a second input terminal of the latch 1105 forms a scan data input terminal 1104, SI. The output terminal of the latch 1107 forms a scan data output terminal, SO, and is also connected to the control terminal 15 of the multiplexer 1108. The output terminal of the latch 1106 forms an output probe path, Q, for the logic analyzer channels.

The scan clock signals, A_clk and B_clk respectively, and the Scan_mode signal configure the PSE 1000. For serial shift operations, the serial input (SI) on the line 20 1104 is captured into the latch 1105 when the A_clk signal is applied and the output of the latch 1105 is captured into the latch 1106 when the B_clk signal is applied. If the Scan_mode signal on the line 1103 is set to a logic 1, the B_clk signal on the line 1102 is also passed through a multiplexer 1109 and an AND gate 1112 to the latch 1107 by a clock signal line 1111. Thus, non-overlapping A_clk and B_clk signals on the clock 25 signal lines 1101 and 1102 respectively clock serial shift operations in the PSE 1000. Signals scanned into the latch 1105 through line 1104 are also scanned into the latch 1107 (and the latch 1106) and the SO output terminal. This completes the programming of the PSE 1000 such that value that has been loaded into the latch 1107 controls input multiplexer 1108 which selects between two input ports 1109 and 1110. Once the PSE 30 1000 has been programmed, the Scan_mode signal on control line 1103 signal is set to and maintained at logic 0 until the PSE 1000 is programmed with a new value. When Scan_mode signal set to logic 0, the PSE 1000 performs its normal data capture function using the clock signal Cf on the line 1001. The Cf clock signals are passed by the multiplexer 1109 to the latch 1106 by a clock signal 1110. The latch 1106 captures the

signals from the latch 1105 and the multiplexer 1108 at the Cf clock rate and passes the signals out to the Q output terminal. The multiplexed-PSEs shown in Figures 12 and 13 build cost efficient logic analyzer channels.

Once enabled, the analysis engine 215 captures new values first into the
5 flip-flops along the OLA channels and subsequently into the buffer memory 218 using trigger signals that have been pre-programmed and implemented as shown in Figures 9c, 9d and 10.

In one mode of operation of the IC 100 shown in Figure 1b, the human engineer may use the diagnostics console 103 to initialize both of the system logic and the
10 SPU 101. In this manner, the SPU 101 may be programmed to perform logic analyzer functions and specific probe points may be enabled so that a history of data values appearing at the selected probe points can be captured by SPU 101. Additionally, the trigger logic shown in Figures 9 and 10 may be programmed to select a desired trigger event in order to stop the data capture operations. Next, the diagnostics console 103
15 invoke the IC 100 to execute its normal system operations. If and when the selected trigger event is detected and the analysis engine 215 has captured the required data, the diagnostics console 103 instructs the SPU 101 to transfer the captured data values out of the IC 100 and into the diagnostics console 103 where the data may be formatted and presented for analysis and interpretation. The diagnostics console 103 and the SPU 101
20 can constrain some of the signals on one or more test wrappers 102 in order to affect the behavior of the IC 100 and perform logic analysis under these conditions. For example, this approach may be useful to determine how the overall behavior of the IC 100 is affected when some of the functionality of any one of the blocks 106 is disabled.

In a different mode of operation automatic test equipment (ATE) may
25 access the IC 100 through its TAP interface 217 in order to initialize the SPU 101 so that internal scan strings 403 and test wrappers 102 are loaded with predetermined test values. The response of the blocks 106 is observed using the scan strings 403 and test wrappers 102. Furthermore, the ATE may be programmed to instruct the SPU 101 to execute BIST or other buffer memory 218 test functions and to check the results to determine pass or
30 fail conditions.

In yet another mode of operation, it is possible to use an in-circuit test (ICT) or similar board-level test equipment to access the IC 100 through its TAP interface 217 in order to instruct the SPU 101 to turn-on its external memory test function. In this mode, patterns are generated by the SPU 101 and made to appear at specific I/O pins of

the IC 100 which are coupled to external memory. For example, the IC 100 may generate the data and address values that are applied to the external memory. The data responses received are capture in order to determine if the external memory is functioning correctly.

- 5 While the description above provides a full and complete disclosure of the preferred embodiments of the present invention, various modifications, alternate constructions, and equivalents will be obvious to those with skill in the art. Thus, the scope of the present invention is limited solely by the metes and bounds of the appended claims.

09275726-032499
664220-92757260